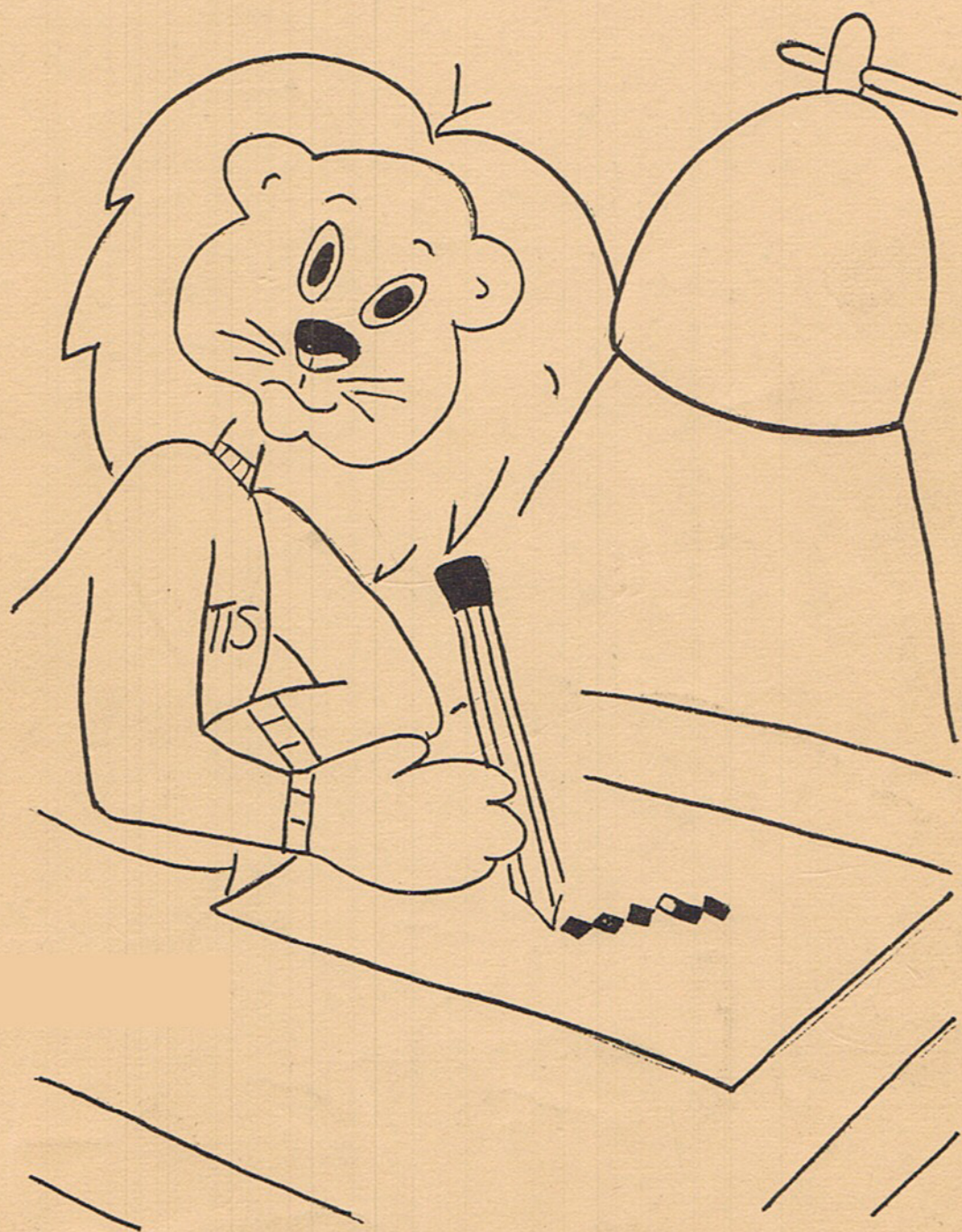




CONTENTS

I.	INTRODUCTION.	1-1
A.	Assumptions Made About the User	1-1
B.	PET Keyboard and TV Display	1-1
C.	Notation Used in This Workbook.	1-2
II.	PET GRAPHICS FEATURES	2-1
A.	Graphics Hardware	2-1
1.	TV Display Refresh Memory	2-1
2.	Character Generator ROM	2-2
3.	Cursor Control Characters	2-3
4.	Using the Cursor.	2-3
B.	Graphics Software	2-5
1.	Character Set Selection	2-5
2.	PRINT Statement	2-6
a.	Standard Character Set.	2-6
b.	Alternate Character Set	2-7
3.	POKE Statement.	2-8
a.	Standard Character Set.	2-9
b.	Alternate Character Set	2-9
4.	The Complete PET Character Set.	2-10
a.	Standard Character Set.	2-10
b.	Alternate Character Set	2-12
III.	LOW-DENSITY PLOTTING.	3-1
A.	Resolution of Low-density Plots	3-1
B.	Histograms and Other Bar Charts	3-1
C.	XY Plots.	3-4
1.	A Screen Coordinate System.	3-5
2.	Conversion from Data to Screen Coordinates.	3-5
D.	Sample Low-density Plots.	3-6
IV.	HIGH-DENSITY PLOTTING	4-1
A.	Resolution of High-density Plots.	4-1
B.	XY Plots.	4-1
1.	Screen Coordinate System Refinements.	4-1
2.	Conversion from Data to Screen Coordinates.	4-1
3.	Organization of the Table of Symbols.	4-2
C.	Sample High-density Plots	4-6
D.	Histograms and Other Bar Charts	4-7
V.	SKETCHING ON THE PET DISPLAY.	5-1
A.	Entering a Sketch	5-1
B.	Moving a Sketch	5-4

VI.	GRAPHICS HINTS AND KINKS.	6-1
A.	When is π not π ?	6-1
B.	What Happened to the Reverse Video?	6-1
VII.	APPENDICES.	A-1
A.	PROGRAM LISTINGS.	A-1
B.	CHARACTER SETS.	B-1

I. INTRODUCTION

This workbook explores some of the graphics capabilities and features of the Commodore PET 2001. In it the PET's memory, TV display characteristics, and its character sets are described. You will be shown how to "draw" histograms and XY plots using both low- and high-density plotting modes. Sketching and moving sketches are also covered.

A. Assumptions Made About the User

Some PET users are comfortable with mathematics. However, this workbook assumes that the majority are not. For that reason, most exercises will use nothing more than high school arithmetic and a small amount of trigonometry.

This workbook also assumes you know something about the syntax and semantics of the BASIC programming language. If you do not, you should obtain one of the following books:

BASIC Programming, J. Kemeny and T. Kurtz
BASIC, Albrecht, Finke, Brown
What Do You Do After You Hit Return?, Peoples Computer Co.
Basic BASIC, James Coan
Advanced BASIC, James Coan

B. PET Keyboard and TV Display

The PET keyboard and display are the primary input and output devices. To successfully communicate with your PET, you must understand the features of the keyboard and how to use the video display effectively. We recommend that you learn this by first doing the exercises in the manual that Commodore supplied with your PET.

Some of the more important characteristics of the PET keyboard input are:

1. For all nongraphics in BASIC, lower case is all that is needed.
2. All lines end with 'RETURN'.
3. The 'DEL' key will delete characters.
4. If there is no keyboard response to the 'STOP' or 'RETURN' keys, power off/on reloads the system and gives a fresh start. Any program entered is lost.
5. To delete the line you are typing, press the 'CLR' key. That deletes the entire line typed so far, and it also erases the entire screen. All previous lines are still in memory.

INTRODUCTION

To use your PET, plug it into a 110 volt AC circuit and turn the power switch on the back of the unit to ON. The first response that you should see on the TV display is:

```
R: ***COMMODORE BASIC***  
R: 7167 BYTES FREE  
R: READY.
```

All numbers listed for free space reflect an 8K RAM system. If you have a different size memory, the number of free bytes will be different.

C. Notation Used in This Workbook

A consistent notation is used in this workbook to indicate what is to be typed on the keyboard (T:), what appears on the TV display (R:), and what indicates blanks are to be typed (b). For example:

```
T: info ('RETURN' key)
```

means to type the characters contained on the line after the colon (:) followed by a 'RETURN'.

```
R: response
```

means that the system response to the previous T: line should be a line on the TV.

Blanks are important. They are specified by b. For example:

```
T: ?"ABbC" ('RETURN' key)
```

means type ?, then ", then the letter A, then the letter B, then a space, then the letter C, then " followed by a 'RETURN'.

Now let's run that example all together:

```
T: ?"ABbC" ('RETURN' key)  
R: AB C
```

The 'RETURN' key must be pressed at the end of each line. We will assume you now know that and will not use ('RETURN' key) in any more examples.

PET GRAPHICS

The special keys on the PET keyboard can cause some confusion. This workbook identifies the special keys with the notation 'KEYNAME'. 'KEYNAME' means press the named key. The unquoted sequence of characters OTHER means press the five keys O T H E R in succession.

Example: Special key notation:

T: 'STOP'
R:

means press the key labelled STOP.

T: STOP
R: BREAK
R: READY.

means press the four keys S T O P in succession.

II. PET GRAPHICS FEATURES

The PET provides graphics output by a combination of hardware and software. The hardware includes the TV display, main memory for refresh, a character generator with a read only memory (ROM), and the keyboard cursor control characters. The software for graphics output is the two BASIC commands PRINT and POKE.

A. Graphics Hardware

The display on the PET is very similar to your standard TV. It is a raster scan device. A dot of light traces out a pattern from left to right and from top to bottom of the screen. By repeatedly blanking and unblanking the light beam, a series of dots can be written on the TV screen.

1. TV Display Refresh Memory. The TV display itself has no way to save (store) information. Since the light dots have a very short life, they must be rewritten frequently. Therefore, the image is rewritten (refreshed) from that portion of the memory that is permanently reserved for data to refresh the TV display. Any information input to this area of memory is automatically displayed on the TV screen. Fig. 2-1 shows a functional diagram of the TV and memory connection. The character generator can be set to select either the standard or the alternate character set.

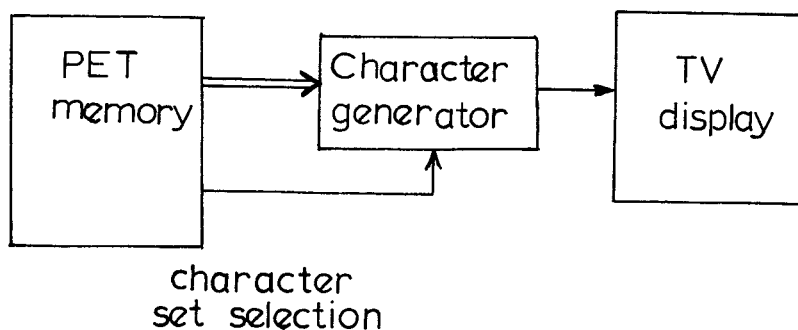


Fig. 2-1. Functional connection of memory and TV.

PET Memory addresses

[illegible]

Fig. 2-2. TV screen/PET memory location correspondence.

The PET uses memory starting at location 32768 for the TV display. The display is organized by row (line). The 40 locations from 32768 through 32807 are the first (top) row of the display. Location 32768 is the left-most character and 32807 is the right-most character.

In order to address any location on the screen, use the formula:

$$\text{LOC} = 32768 + 40 * (\text{ROW}-1) + (\text{COL}-1)$$

where ROW is the number of the row (1-25) and COL is the number of the column (1-40).

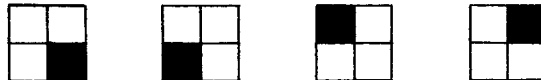
The PET is an 8-bit machine. This means that each byte (location) in the PET can hold a number from 0 to 255. This means there are 256 possible values, each of which has a symbol associated with it.

2. Character Generator ROM. All characters are displayed on an 8x8 dot matrix. The characters in the PET main memory are used to select the pattern of dots to be turned on and off in the 8x8 grid. The dot patterns are in the character generator read only memory.

PET GRAPHICS FEATURES

Even though the TV displays 320 (8x40) horizontal dots and 200 (8x25) vertical dots, you cannot display 320x200 distinct points. The character generator ROM limits the combination of dots that can be displayed. See Fig. 2-1 and note that two different character sets can be displayed.

The PET display has 25 lines of 40 characters per line. If each character is considered a single point, a total of 1000 points can be displayed. Some of the "characters" can provide four times the graphical information as others. For example, the symbols



double the resolution in each direction, so you have 50 lines of 80 points. In this high-density mode, a total of 4000 points can be displayed.

3. Cursor Control Characters. The PET keyboard includes 7 cursor control characters. The most commonly used cursor control character is the 'RETURN'. This character causes the cursor (the symbol which shows where the next characters will go) to go down one line and move to the extreme left.

There are two cursor control characters which move the cursor to the upper left-hand corner of the screen. The 'HOME' character does not change the contents of the screen. The 'CLR' character puts blanks in all screen positions before moving the cursor to the upper left-hand corner of the screen.

The other four cursor control characters move the cursor one position up, down, left, or right.

4. Using the Cursor. A character can be printed at any point on the screen by the proper selection of cursor control characters followed by the character you want to display. The cursor can be moved one location in any of the four directions: up, down, left, and right. The 'CLR' and 'HOME' characters move the cursor to the upper left-hand corner of the screen.

Exercise: The following program randomly moves a character around the screen using cursor control.

```
T: NEW
T: 100 PRINT "'CLR'"
T: 102 FOR I=1 TO 8
T: 104 READ PS$(I)
T: 106 DATA "↑", "→", "↓", "←", " ", " ", " ", " "
T: 108 NEXT I
```

PET GRAPHICS

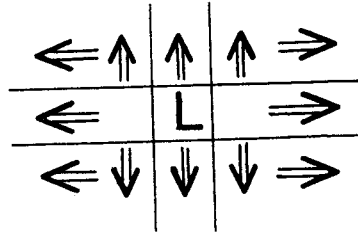


Fig. 2-3. Cursor control characters to move to the eight adjacent locations.

Clear the TV screen and then load the character array PS\$ with the eight possible moves you can make from any non-edge location on the screen. Fig. 2-3 shows the eight locations and the cursor control characters needed to get to each of these locations from the original position.

```
T: 110 P%=RND(1)*8+1
T: 120 FOR I=1 TO 10
T: 130 PRINT " ← b";PS$(P%);"#";
T: 135 NEXT I
T: 140 GO TO 110
```

Pick a random direction to move in. Move ten steps in the selected direction by backing up over the # symbol and writing over it with a blank. Then move across the screen by printing the cursor control symbol(s) and the # symbol. After ten steps, go back for another random direction.

```
T: RUN
```

The # symbol should bounce around the screen in a random fashion. To stop the program type:

```
T: 'STOP'
R: BREAK IN 130
R: READY.
```

In this case, the line number is 130. The line number will depend on when you press the 'STOP' key.

PET GRAPHICS FEATURES

B. Graphics Software

The software on the PET allows you to use the TV display as a "glass teletypewriter". Lines of information can be displayed just as they would on a teletypewriter. When more information is displayed than will fit on the TV screen, the oldest information is rolled (scrolled) off the top of the screen and lost forever. Besides displaying alphanumeric characters, the PET can display graphical symbols.

1. Character Set Selection. In the section on graphics hardware we learned that the PET can display an 8x8 dot matrix character. (See Fig. 2-4.) The dots that are on and off are determined by the character generator ROM. Actually the character generator ROM has two possible sets of 8x8 dot patterns. One set, let's call it the standard set, displays the upper case letters, numbers, special characters, and graphical symbols. The second set, let's call it the alternate set, displays upper-case letters, numbers, special characters, and lower-case letters. You cannot display both lower-case letters and graphical symbols at the same time.

You can tell your PET which character set you want to use by setting one of two specific values (12 or 14) in a particular memory location (59468). The character generator uses this information to select the dots to blank and unblank for a given 8-bit value. When you turn power on, the standard (graphical) character set is automatically selected.

Exercise: Switch to the alternate character set and display some lower-case letters.

T: POKE 59468,14

Type the 'SHIFT' key and a few characters. Lower-case letters are displayed instead of graphics symbols, so it is necessary to shift to specify lower case letters.

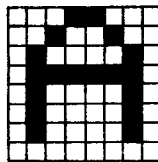


Fig. 2-4. An example of an 8x8 matrix showing the character A.

PET GRAPHICS

T:
T:
T:

Exercise: Switch back to the standard (graphical symbols) character set.

T: 'CLR'
T: POKE 59468,12

Try typing the same shifted characters.

T:
T:
T:

2. PRINT Statement. The PRINT statement displays some characters and treats others as cursor control so that not all 256 characters are visible on the screen. You can use either the standard or alternate character set with the PRINT statement.

- a. Standard Character Set. First of all, do the following exercise to determine what characters can be displayed with PRINT and what the other special characters do.

The following program displays each of the possible characters along with its numeric value (0-255). After each pair is printed, the program waits for a response (press any key) from the keyboard.

Exercise: PRINT all 256 possible characters.

```
T: NEW
T: 100 FOR I=0 TO 255
T: 110 PRINT CHR$(I),I
T: 120 GET A$:IF A$=""THEN 120
T: 130 NEXT I
T: RUN
R: 0
```

The blanks followed by zero indicate that no symbol is displayed for the character whose numeric value is 0.

PET GRAPHICS FEATURES

T: any key
R:

1

Repeat typing to get a new line until

R: 12

Carefully type any character.

T: any key
R:
R:

13

The blank line shows that the character whose numeric value is 12 is a line feed.

Continue typing any key until

R: π 255
R: READY.

Note that you cannot PRINT any reverse video (black on white instead of white on black) characters. You can PRINT cursor control, graphics and, of course, capital letters, numbers, and special mathematical symbols.

b. Alternate Character Set.

Exercise: Experiment with the alternate character set. Select the alternate character set by storing (POKE) the number 14 in the special location associated with the character generator.

T: POKE 59468,14
T: RUN
R: 0

Continue typing any key until

R:  255

PET GRAPHICS

Notice that lower case alphabetic characters are displayed instead of some of the graphic symbols. To return to the standard character set:

T: POKE 59468,12
R: READY.

3. POKE Statement The second method of displaying information is to put it directly into the memory used to refresh the TV. To do this, you must use the POKE statement. POKE requires a memory address and a numerical value to store at that address. The BASIC function ASC converts a character to a number.

Try the following exercise for direct display.

Exercise: Direct display on the TV.

T: 'CLR'
T: POKE 32768,ASC("#")

This will display the character # in the upper left hand corner of the TV screen.

Exercise: Randomly fills the screen with the specified character.

T: NEW
T: 400 PRINT "'CLR'"
T: 410 D%=RND(1)*1000+1
T: 420 POKE 32767+D%,ASC(".")
T: 430 GO TO 410
T: RUN

To end this program:

T: 'STOP'

It is best to then

T: 'CLR'

This eliminates any problems with characters left on the screen being confused with commands you subsequently type.

PET GRAPHICS FEATURES

a. Standard Character Set.

Exercise: POKE-Standard character set.

Change program line 420 to an alphabetic character by:

```
T: 420 POKE 32767+D%,ASC("A")
T: RUN
```

The spade (♠) symbol will be displayed.

```
T: 'STOP'
T: 'CLR'
T: 420 POKE 32767+D%,ASC("A")-64
T: RUN
```

The upper-case A will be displayed.

```
T: 'STOP'
T: 'CLR'
T: 420 POKE 32767+D%,ASC("A")+64
R: RUN
```

The reverse video upper-case A will be displayed.

b. Alternate Character Set.

Exercise: POKE-Alternate character set.

Change to the alternate character set:

```
T: 'STOP'
T: 'CLR'
T: POKE 59468,14
T: RUN
```

Repeat the previous examples using `ASC("A")+64`, `ASC("A")-64`, and `ASC("A")`. The only case that should be different is the third one. Lower-case a's should be displayed (instead of the spade).

PET GRAPHICS

To return to the graphics character set:

```
T: 'STOP'
T: 'CLR'
T: POKE 59468,12
```

4. The Complete PET Character Set. To display all 256 possible characters, we will POKE the numeric value of the character to the TV refresh portion of the PET memory. All values will be used so that the value of a particular character should not concern you now.

- a. Standard Character Set.

Exercise: Set up a program to display all the possible PET characters.

Part 1. Specify graphical characters or lower-case alphabetic characters and generate the table of symbols.

```
T: NEW
T: 90 PRINT "GRAPHICS OR LOWER CASE"
T: 92 INPUT A$
T: 94 IF A$="GRAPHICS" THEN POKE 59468,12
T: 96 IF A$="LOWER" THEN POKE 59468,14
T: 99 PRINT "'CLR'"
```

Prompt for character set selection. Ignore any unrecognized response. Clear the screen for action.

```
T: 100 PRINT "CHARACTER DISPLAY TABLE (POKE)"
T: 110 PRINT "    00          10          20"
T: 120 PRINT "    01234567890123456789012345"
T: 125 Q=32891
T: 130 FOR J=0 TO 9
T: 140 PRINT J;SPC(26);J
T: 150 NEXT J
T: 152 PRINT
T: 154 PRINT "    00          10          20"
T: 156 PRINT "    01234567890123456789012345"
```

Put a label and the character position information on the screen. The symbols are displayed in a table so that the numerical equivalent of a character can be determined by taking ten times the column number plus the row number.

PET GRAPHICS FEATURES

```
T: 160 FOR J=0 TO 9
T: 170 FOR I=0 TO 24
T: 180 POKE Q+((J+1)*40+I),I*10+J
T: 190 NEXT I
T: 210 NEXT J
```

Put ten rows (0-9) on the screen for all 24 columns.
Calculate the characters numeric value as ten times the column number plus the row number.

```
T: 215 I=25
T: 220 FOR J=0 TO 5
T: 230 POKE Q+((J+1)*40+I),I*10+J
T: 240 NEXT J
T: 250 STOP
```

geeft Break in 250, want door je weet waar je verder moet. (cont 300)

The 25th column is a special case since it has only six rows (0-5).

Part 2. Input a numeric value and display the corresponding character in the upper left-hand corner. This can be used to double check the value of a character.

```
T: 300 INPUT K
T: 305 IF K < 0 OR K > 255 THEN 315
T: 310 POKE 32848,K
T: 315 PRINT "↑↑↑"
T: 320 GO TO 300
```

This is a separate program which accepts a numeric value between 0 and 255 and POKES it to a fixed location on the TV display. You can use it to check the value of specific characters. The "↑↑↑" in line 315 keeps the input line in the same spot so that the character table does not scroll off the screen.

```
T: RUN
R: GRAPHICS OR LOWER CASE
R: ?
T: GRAPHICS
```

*↓ waar schijnt de
1 van wege opschrijven vraagtekens
1 van wege print opdraaiing*

The table of graphics symbols and upper-case letters will be displayed.

PET GRAPHICS

T: RUN 300
R: ?
T: 0

The character corresponding to the number typed will be displayed under the C in the heading. Continue to type the number of the character you are interested in. Be sure to completely overwrite any previous number. If you type any illegal character number, it will be ignored.

T:
T:
T:
T: 'STOP'

This terminates the number to symbol display program.

b. Alternate Character Set.

T: RUN
R: GRAPHICS OR LOWER CASE
R: ?
T: LOWER

The table of lower-case letters and upper-case letters will be displayed.

T: RUN 300
R: ?
T: 0

The character displayed will be from the upper/lower-case character set.

T:
T:
T:
T: 'STOP'

This terminates the number to symbol display program.

III. LOW-DENSITY PLOTTING

Low-density plots are useful to show gross differences. Using characters to draw lines and curves can give you a rough picture of a relationship. Other approaches need to be developed to show fine differences.

A. Resolution of Low-density Plots

The PET display has 25 lines of 40 characters each. If we use each character position to display either a "point" or a blank, we can display 1000 points on the TV screen. With so few points, it is difficult to draw lines with much accuracy. Lines drawn in low-density mode can have blank spaces between points. These lines can give an approximate but not a very exact picture.

B. Histograms and Other Bar Charts

We can use low-density plotting to draw histograms and bar charts. A histogram is a graphical representation of a frequency distribution in which the heights of vertical bars are proportional to the class frequency and each bar represents a class. See Fig. 3-1, for an example. A general bar chart uses the length of the bar to represent a characteristic and the location of the bar represents the variable with that characteristic. See Fig. 3-2, for an example.

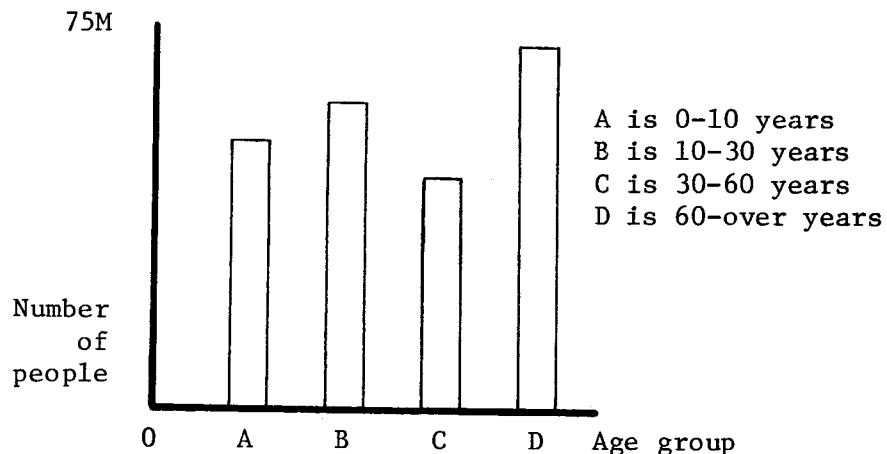


Fig. 3-1. Histograms of population.

PET GRAPHICS

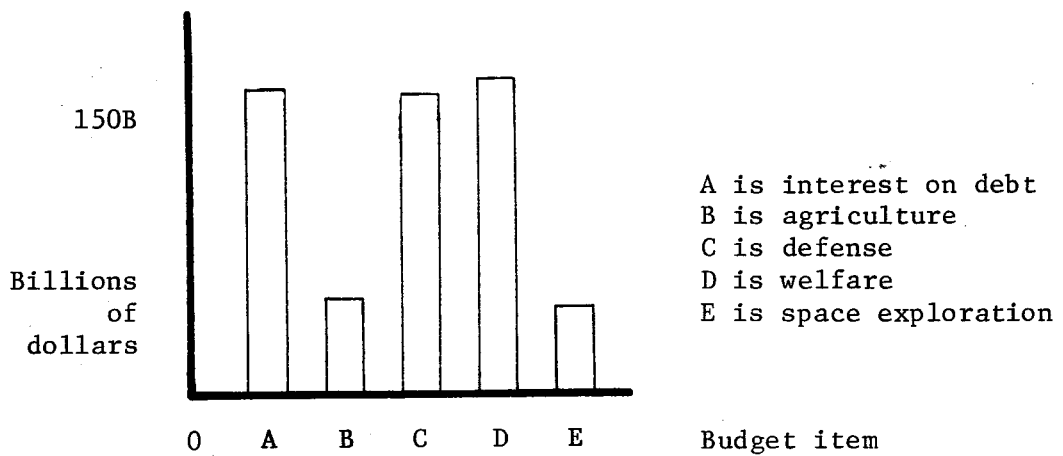


Fig. 3-2. Budget breakdown.

Neither figure has any basis in fact. The examples are there to give you the idea of how effectively relationships can be shown with graphics.

Bar graphs and histograms are simpler to generate than general XY plots. Normally the range of the X variable is divided into intervals. The value of y corresponding to an x determines how tall the column is.

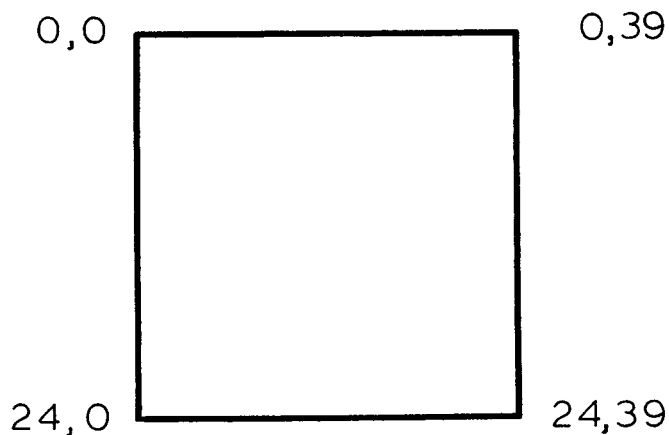


Fig. 3-3. TV display X,Y coordinates.

LOW-DENSITY PLOTTING

Each X (character) position can contain a vertical bar. However, to make the chart more readable, a blank column will be inserted between each vertical bar. This leaves a total of 20 vertical bars maximum. The vertical resolution is one space in 25 or approximately 4%. Any two vertical bars that differ by less than 4% will have the same height. In order to emphasize the differences, the lower (common) part of the bar chart is often deleted.

Example: Assume that the shortest vertical bar in Fig. 3-1 (C group) is just over 45 million. If we start the graph at 40 million instead of zero, we can double the resolution and make any differences more visible. Fig. 3-4 shows the effect of scaling.

Exercise: Histogram plot of the randomness of RND.

```
T: NEW
T: 100 REM HISTOGRAM PLOT
T: 102 DIM Y%(20)
T: 120 BL%=ASC("█")-64
T: 130 FOR I=1 TO 20
T: 140 Y%(I)=0
T: 150 NEXT I
```

See spaces

*teny weker 0% als int-functie. dit bl 5
re bl 2 3-5-*

Initialize the counters and the character to plot.

verhogen

```
T: 160 FOR I=1 TO 200
T: 170 IN%=RND(1)*20+1
T: 180 Y%(IN%)=Y%(IN%)+1
T: 190 NEXT I
```

*Dit programma geeft een verdeling bij
een klein tussen 1 en 20*

IN% tussen 1 en 20

Generate 200 random numbers and count the occurrences in each interval.

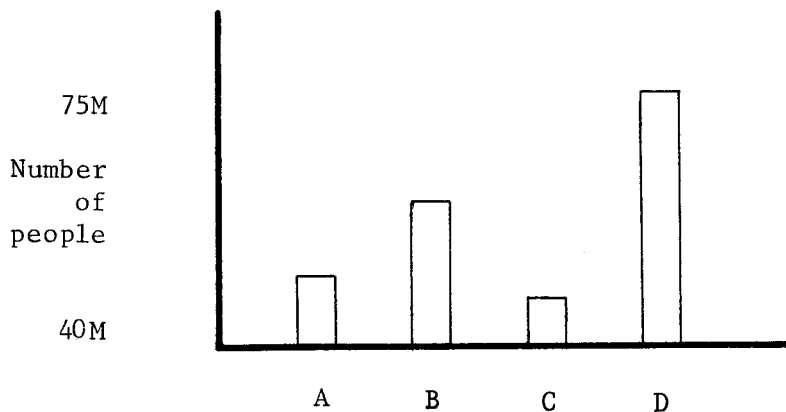


Fig. 3-4. Scaled histogram of population.

PET GRAPHICS

```

T: 195 PRINT "'CLR' HISTOGRAM PLOT OF RND"
T: 200 FOR I=1 TO 20
T: 210 X%=I*2-1
T: 220 FOR J=0 TO Y%(I)
T: 230 POKE 32768+((23-J)*40+X%),BL%
T: 240 NEXT J
T: 250 NEXT I
T: 260 GO TO 120

```

Handwritten notes:
 → maakt staven in kolommen 1 t/m 39 met daartussen lege kolom.
 ← schrijft staaf voor staaf.

Display the number of occurrences in each interval. Then go back for more.

T: RUN

A histogram will be repeatedly plotted. To stop the program:

T: 'STOP'

C. XY Plots

For most plotting applications the X and Y coordinates and the data item to be plotted at that point are calculated. To make the PET TV display into an X Y plot output device we have to map (convert) from the X Y coordinate system the data is in to the PET TV address space. In order to define the meaning, the user must specify the upper and lower limits for both X and Y. The four numbers Xmin, Xmax, Ymin, and Ymax are needed to define the user's plotting surface and coordinate system as shown in Fig. 3-5.

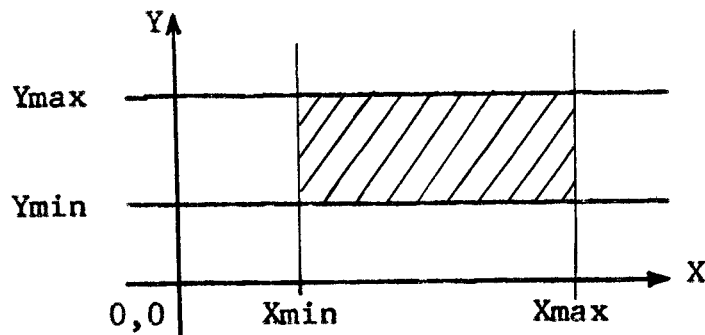


Fig. 3-5. User Coordinate System.

LOW-DENSITY PLOTTING

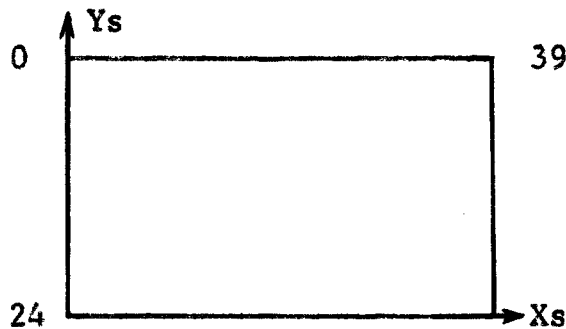


Fig. 3-6. Screen Coordinate System.

1. A Screen Coordinate System. Let's introduce a screen coordinate system as shown in Fig. 3-6, where Ys is an integer increasing from 0 to 24 when we move down the screen; Xs is an integer increasing from 0 to 39 when we move from left to right.
2. Conversion from Data to Screen Coordinates. The conversion from data coordinates (X,Y) to screen coordinates (Xs,Ys) can be done in two steps. First, calculate the line and character position on the TV screen that corresponds to Xs, Ys. Then calculate the memory location that corresponds to the position on the screen.

Exercise: Build a plot subroutine.

T: NEW
T: 1000 XS%=39*((X-XL)/(XB-XL)) *39 is de factor die de lengte van interval (XL, XB) afbeeldt op de lengte van een karakter*
T: 1010 YS%=24*((YB-Y)/(YB-YL)) *Analooog 24 voor een rij.*
Vgl blz 4-1.

The symbol % means that the variable is an integer.

XL is the minimum value of X (Xmin)
XB is the maximum value of X (Xmax)
YL is the minimum value of Y (Ymin)
YB is the maximum value of Y (Ymax)

YS% is the TV line number and XS% is the TV character number; that is, the coordinates of the particular point on the screen that corresponds to (X,Y).

T: 1050 Q=YS%*40+XS%+32768
T: 1310 POKE Q,PT%
T: 1320 RETURN

PET GRAPHICS

The value of Q is the location in the memory that corresponds to line YS% and character position XS%. Do not be concerned about the strange choice of statement numbers. Later, the gaps will be filled in as we make this subroutine more powerful.

D. Sample Low-density Plots

To display a symbol at the calculated point on the screen you must store (POKE) the symbol in the corresponding memory location.

Exercise: Make a low resolution X Y plot of $Y=\sin(X)$ using the subroutine you just built.

T: 100 PT%=ASC("*") *→ problem with (".") = (".*") + 4?*
T: 103 PRINT "'CLR' PLOT OF SINE 0 TO 2π "
T: 104 XL=0:XB= $2*\pi$
T: 105 YL=-1.2:YB=1.2
T: 106 SS=.1
T: 107 FOR I=XL TO XB STEP SS
T: 108 X=I
T: 110 Y=SIN(X)
T: 120 GOSUB 1000
T: 130 NEXT I
T: 140 GO TO 103
T: RUN

A repeated plot of $\sin(X)$ will be displayed. To terminate the program:

T: 'STOP'
R: BREAK IN 110
R: READY.
T: 'CLR'

IV. HIGH-DENSITY PLOTTING

High-density plots can show slightly more information than low-density plots. With more points displayed, a somewhat smoother picture is drawn.

A. Resolution of High-density Plots

The PET display has 25 lines of 40 characters. If we use each character position to display four points, we can display 4000 different points. The PET graphical character set includes the symbols:



We will call these symbols "quarter blocks". With these extra points we can draw lines with more accuracy than in the low-density mode.

B. X Y Plots

The same approach can be used for a high-density plot as for a low-density plot. The process must be extended to determine which of the four possible positions that the quarter block has.

1. Screen Coordinate System Refinements. The mapping of data (X,Y) to screen (Xs, Ys) uses the same basic formulas to determine line and character position. An adjustment must be made for the higher density.

T: 1000 $XS\% = 79 * ((X - XL) / (XB - XL))$
T: 1010 $YS\% = 49 * ((YB - Y) / (YB - YL))$

vgl. blz 3-5.

020

These formulas reflect the fact that there are 80 horizontal and 50 vertical positions.

2. Conversion from Data to Screen Coordinates. The YS% must be transformed to a line number (YL%) and XS% to a character number (XC%).

T: 1020 $XC\% = XS\% / 2 ; YL\% = YS\% / 2$
T: 1050 $Q = YL\% * 40 + XC\% + 32768$

These statements give the memory location corresponding to the row and character position needed.

PET GRAPHICS

3. Organization of the Table of Symbols. The next step is to determine which block is needed. There are four choices. First, determine if a left or right block is needed with the formula:

T: 1060 $I\% = XS\% \text{ AND } 1$? 

$I\%$ will be a zero if a left block is needed and a 1 if a right block is needed. Then, determine if an upper or a lower block is needed with the formula:

T: 1062 $J\% = (YS\% + 1) \text{ AND } 1$

$J\%$ will be a zero if a lower block is needed and 1 if an upper block is needed.

By the proper choice of data array and the formula

T: 1064 $K\% = (J\% + 1) * (I\% * 3 + 1)$

the desired block will be selected.

Example: High density point plotting data array organization.

Consider the four sets of screen coordinates ($XS\%$, $YS\%$):

0,0; 0,1; 1,0; 1,1

Using the formulas from 1060, 1062 and 1064, verify that $K\%$ for

0,0 is 1
0,1 is 2
1,0 is 4
1,1 is 8

If we set

TB(1) = "  "
TB(2) = "  "
TB(4) = "  "
TB(8) = "  "

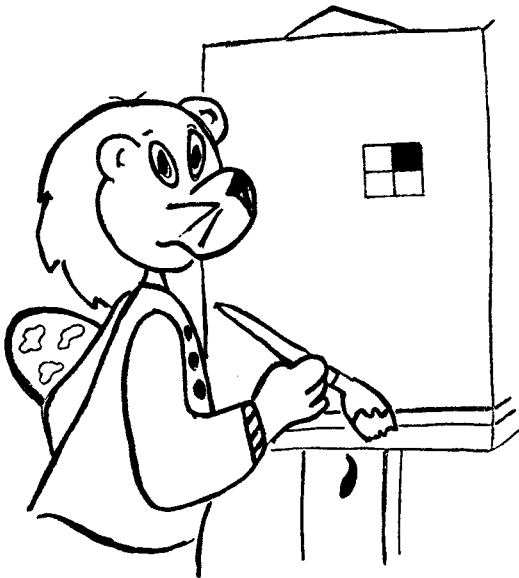
the subscript calculated gives the proper quarter block.

HIGH-DENSITY PLOTTING

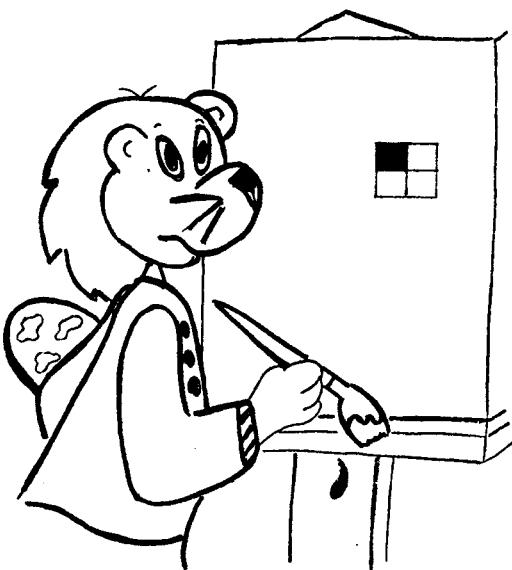
If we put the proper block on the screen with

```
T: 1308 PT%=TB(K%)  
T: 1310 POKE Q,PT%
```

the three neighboring blocks would be erased. We must organize the rest of TB so that it is easy to combine the information that is already plotted on the screen with the information.

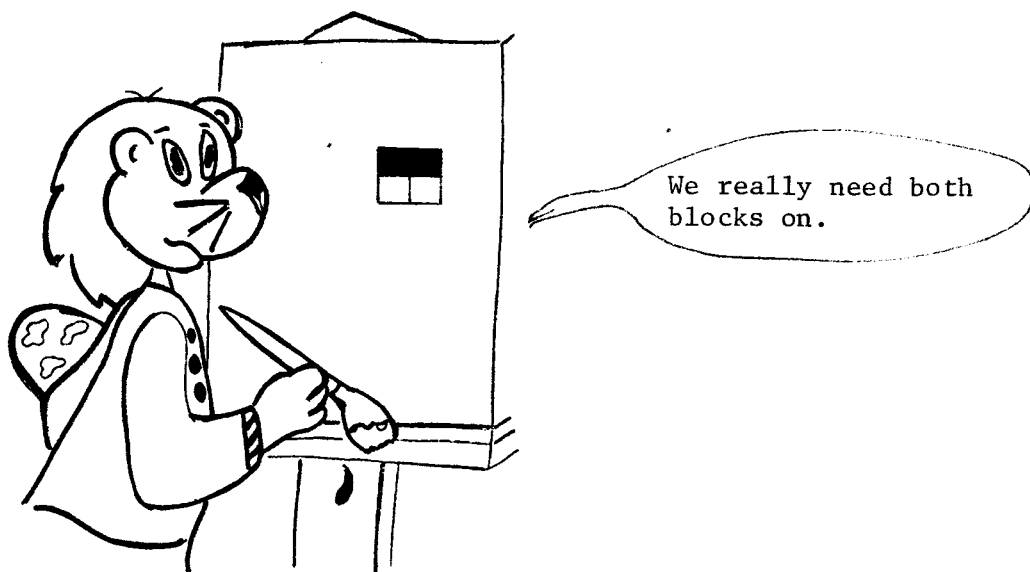


We can start with a point turned on.



If we just turn on the next point, we lose the old information.

PET GRAPHICS



Since the subscript identifies the character to display, the remaining entries in TB must be organized so that the subscript/character to be displayed gives the desired combination. This means that we want an OR function of the subscript, not a numerical add.

Exercise: Fill in the rest of the high-density point plotting table. We already have the building blocks TB(1), TB(2), TB(4), and TB(8). Consider all the possible combinations of quarter blocks displayed:

NOTE: <i>all reverse video</i> NOTE:		is TB(1) or TB(2) = TB(3)
		is TB(1) or TB(4) = TB(5)
		is TB(2) or TB(4) = TB(6)
		is TB(5) or TB(2) = TB(7)
		is reverse video of
		is TB(8) or TB(1) = TB(9)
		is TB(8) or TB(2) = TB(10)
		is TB(10) or TB(1) = TB(11)
		is TB(8) or TB(4) = TB(12)
		is TB(12) or TB(1) = TB(13)
		is TB(12) or TB(2) = TB(14)
		is TB(14) or TB(1) = TB(15)
		is reverse video of blank (space)

HIGH-DENSITY PLOTTING

Since zero subscripts are permitted and a blank means zero (no) blocks on

$$\square = \text{TB}(0) \quad \text{so } \text{TB}(0) = "b"$$

Many of these characters in TB cannot be entered easily since some of them are reverse video. The numerical equivalent of these characters is entered with

```
T: 90 DIM TB(15)
T: 91 FOR I=0 TO 15
T: 92 READ TB(I)
T: 94 DATA 32, 123, 126, 97, 108, 98, 127, 252, 124, 255 BL B5
35, 226, 236, 225, 254, 251, 160
T: 96 NEXT I
```

If we look at what is already in the memory to determine which blocks are already "on" we can OR the current block to whatever was on the screen before.

The sequence of code

```
T: 1070 K=PEEK(Q)
T: 1080 FOR J=0 TO 15
T: 1090 IF K=TB(J) THEN 1300
T: 1100 NEXT J
T: 1110 J=0
```

takes the information that was previously on the screen and searches for a match in the table of possible quarter blocks. The 'CLR' command originally set the entire screen to blanks (character code 32). The table is organized so that the index of the character found on the screen when "ORed" with the index of the new quarter block to be displayed will yield the index of the character created by "ORing" the characters together. If an unrecognized block is found, it is treated as a blank.

```
T: 1300 K%=K% OR J
T: 100
T: RUN
```

venwage blz 3.6.

You should get a sine plot. Vary the step size (SS) in line 106 to improve the smoothness of the curve.

doer

PET GRAPHICS

C. Sample High Density Plots

Functions of two variables.

Exercise: Plot a straight line.

```
T: 103 PRINT "'CLR' Y=F(X)"
T: 108 X=I
T: 110 Y=X/6
T: RUN
R:
```

A straight line will be plotted.

Exercise: Move Y intercept to the straight line.

```
T: 110 Y=X/6-1
T: RUN
R:
```

The same straight line will be plotted lower on the screen.

Exercise: Change the slope of the straight line.

```
T: 110 Y=X/3-1
T: RUN
R:
```

A steeper straight line will be plotted.

Exercise: Plot $Y = \cos(x)$ from 0 to 2π .

```
T: 110 Y=COS(X)
T: RUN
R:
```

A cosine curve will be plotted starting at $(0, \cos(0))$.

Exercise: Plot $Y = \cos(x)$ from $-\pi$ to $+\pi$.

```
T: 104 XL=-PI:XB=PI
T: RUN
R:
```

A cosine curve will be plotted starting at $(-\pi, \cos(-\pi))$.

HIGH-DENSITY PLOTTING

Exercise: Change the screen mapping limits and plot $Y=\text{SQR}(X)$.

```
T: 104 XL=0 XB=1000
T: 105 YL=0:YB=35
T: 106 SS=10
T: 110 Y=SQR(X)
T: RUN
R:
```

A square root curve will be plotted. Notice the discontinuity at $X=0$, $Y=0$.

D. Histograms and Other Bar Charts

The PET graphics symbols  and  permit high-density bar charts to be drawn in much the same way as the quarter blocks permit high-density point plots to be drawn.

The half block symbol  allows twice as many X positions to be used. The quarter block  doubles the resolution in the Y direction.

Exercise: Display a high-density histogram of the randomness of RND.

```
T: NEW
T: 100 REM THIN HISTOGRAM
T: 102 DIM Y%(40)
T: 120 BL%=ASC("  ")-64
T: 130 FOR I=0 TO 39
T: 140 Y%(I)=0
T: 150 NEXT I
T: 160 FOR I=1 TO 400
T: 170 IN%=RND(1)*40
T: 180 Y%(IN%)=Y%(IN%)+1
T: 190 NEXT I
```

Count occurrences of RND in each interval.

graph plot

```
T: 195 PRINT "'CLR'HISTOGRAM PLOT OF RND"
T: 200 FOR I=0 TO 39
T: 220 FOR J=0 TO Y%(I)
T: 225 IF Y%(I)=0 THEN 250
T: 230 POKE 32768+((23-J)*40+I),BL%
T: 240 NEXT J
T: 250 NEXT I
T: 260 GO TO 120
T: RUN
```

PET GRAPHICS

A high-density histogram of the randomness of RND will be repeatedly displayed. To terminate the plotting:

```
T: 'STOP'
R: BREAK IN 200 (or some other line)
R: READY.
T: 'CLR'
```

Exercise: Draw a set of stair step bars to demonstrate a high-resolution bar chart.

```
T: NEW
T: 100 REM HI RESOLUTION DEMO
T: 102 DIM Y%(40)
T: 120 BL%=ASC("█")-64
T: 125 B2%=ASC("▒")-64
T: 130 FOR I=0 TO 39
T: 140 Y%(I)=I
T: 150 NEXT I
T: 195 PRINT "'CLR' DEMO OF HI-RESOLUTION BAR CHART"
T: 200 FOR I=0 TO 39
T: 205 J=0
T: 210 IF Y%(I) < 2 THEN 245
T: 220 FOR J=0 TO Y%(I)/2-1
T: 225 IF (INT(Y%(I)/2))=0 THEN 240
T: 230 POKE 32768+((23-J)*40+I),BL%
T: 240 NEXT J
T: 245 IF (Y%(I)AND 1) < > 0 THEN POKE 32768+((23-J)
    *40+I),B2%
T: 250 NEXT I
T: 260 GO TO 120
T: RUN
R:
```

A 40-column bar chart with 0-48 vertical resolution will be repeatedly displayed. To terminate the plotting:

```
T: 'STOP'
R: BREAK IN 220 (or some other line)
R: READY.
T: 'CLR'
```


V. SKETCHING ON THE PET DISPLAY

One way to use the PET graphics symbols is to sketch the desired picture on the screen and then save the resulting picture. The screen when cleared is filled with blanks. Therefore, any non-blank character in the TV display memory area must be something entered as part of the desired picture.



A. Entering a Sketch

The following program allows entry of up to 255 characters. When the symbol ↑ is typed, the picture is finished and saved in a memory area. The program then asks what you want done with the picture. You can either write it on a cassette or redisplay it.

A complete sequential list of this program without explanatory text is included in the appendix.

Exercise: A TV sketch program.

```
T: NEW
T: 100 DIM PIC%(255),LOC(255)
```

Set aside the maximum array size for picture elements. PIC% contains the symbol and LOC contains the TV memory address of the corresponding symbol.

PET GRAPHICS

```
T: 110 A$=""CLR' ↓ ↓ ↓ ↓ "
```

```
T: 115 PRINT "← b ← ";A$;"#";
```

veegt cursor # uit. zorgt er voor dat symbool op plaats van cursor komt

Initialize the character to display and clear the screen. Back up over the # (cursor) and display the character just typed.

```
T: 120 GET A$
```

```
T: 130 IF LEN(A$)=0 THEN 120
```

```
T: 140 IF A$ < > "↑" THEN 115
```

Wait for a character to be typed at the keyboard. Recycle for more characters until the end signal is typed.

```
T: 150 K%=0
```

```
T: 155 PRINT "← b" veegt # uit ; anders zou deze ook in de tekening komen
```

```
T: 160 FOR I=32768 TO 33767
```

```
T: 170 K=PEEK(I)
```

```
T: 180 IF K=32 THEN 220
```

Initialize the pointer to save the image. Wipe out the cursor (#). Look at each location in the TV display memory. If it is non-blank, save it.

```
T: 190 K%=K%+1
```

```
T: 200 IF K% > 255 THEN 900
```

Bump the save pointer to the next location. If the array is exhausted, write an error message.

```
T: 210 PIC%(K%)=K
```

```
T: 215 LOC(K%)=I
```

Save the character on the screen and the location of that character.

```
T: 220 NEXT I
```

```
T: 225 PRINT "'HOME'"
```

```
T: 230 PRINT "WHAT NEXT?"
```

```
T: 240 INPUT A$
```

Prompt for the input which specifies the next action to take.

SKETCHING ON THE PET DISPLAY

```
T: 250 IF A$="W" THEN 400
T: 255 IF A$="N" THEN 110
T: 257 IF A$="R" THEN 500
```

Possible actions are W for write, N for next, D for display, or R for read.

```
T: 260 IF A$ < > "D" THEN 225
```

If action request is not recognized, skip back to ask for input again.

```
T: 270 PRINT "'CLR'"
T: 280 FOR I=1 TO K%
T: 290 POKE LOC(I),PIC%(I)
T: 300 NEXT I
```

Clear the screen and display sketch just saved. Put all the non-blank characters back on the TV screen in the same spot they were found.

```
T: 310 GO TO 230
```

Go back for the next action.

```
T: 400 OPEN 1,1,2,"PICTURE"
T: 405 FOR I=1 TO K%
T: 410 T=TI
T: 420 PRINT #1,STR$(LOC(I));", ";STR$(PIC%(I))
T: 430 IF TI-T < 120 THEN 470
T: 440 POKE 59411,53:T=TI
T: 450 IF TI-T < 6 THEN 450
T: 460 POKE 59411,61
T: 470 NEXT I
T: 475 PRINT #1,STR(0);", ";STR(0)
T: 476 CLOSE 1
T: 478 PRINT "'HOME' WRITE DONE 'HOME'"
T: 480 GO TO 230
```

Write out the character and memory locations. Then go back for the next action.

PET GRAPHICS

```

T: 500 OPEN 1,1,0,"PICTURE"
T: 510 PRINT "'CLR' ↓↓↓↓ " pet
T: 520 INPUT #1,A,B
T: 530 IF A=0 AND B=0 THEN 560
T: 540 POKE A,B
T: 550 GO TO 520
T: 560 CLOSE 1
T: 565 A$="b"
T: 570 GO TO 120

```

Read in data and put it on the screen.

```

T: 900 PRINT "TOO MANY SYMBOLS"
T: 910 K%=255:GO TO 230

```

Print an error message if sketch is too detailed. Save what you can.

T: RUN

B. Moving a Sketch on the TV Display

The screen is mapped to memory locations so that the following relationships hold. Given a point on the screen (not on an edge), call its memory location L. The eight addresses of the adjacent screen locations in memory are:

L-41	L-40	L-39
L-1	L	L+1
L+39	L+40	L+41

Fig. 5-3. Relative Screen Positions.

SKETCHING ON THE PET DISPLAY

If we name the positions:

- UL (up left)
- U (up)
- UR (up right)
- R (right)
- DR (downright)
- D (down)
- DL (down left)
- L (left)
- B (back to original location)

We can use the following program to move any sketch around the original screen location.

Exercise: Move any sketch around the original screen location.

```
T: NEW
T: 100 FOR I=1 TO 9
T: 110 READ N$(I),RL%(I)
T: 120 DATA "UL",-41,"U",-40,"UR",-39,"R",1,
T:      ,"DR",41,"D",40,"DL",39,"L",-1,"B",0
T: 130 NEXT I
```

Set up the name of the move request and the relative displacement from the original screen location.

```
T: 140 FOR I=1 TO 4
T: 150 READ LOC(I),PIC%(I)
T: 160 DATA 33267, 79, 33268, 80, 33307, 76, 3330, 8, 122
T: 170 NEXT I
```

Set up the location and the graphics character to draw a simple four-element picture.

```
T: 180 K=0
T: 185 PRINT "'CLR'"
```

Initialize displacement to none; i.e., the original picture location. Clear the screen.

```
T: 190 FOR I=1 TO 4
T: 200 POKE LOC(I)+K,PIC%(I)
T: 210 NEXT I
```

PET GRAPHICS

Put the picture characters on the screen with the specified displacement. Initially the displacement is zero and the picture is displayed at its original position.

```
T: 220 INPUT DR$
T: 230 FOR I=1 TO 9
T: 240 IF DR$=N$(I) THEN 270
T: 250 NEXT I
```

Get the direction to move from the keyboard and search the table of directions for a match.

```
T: 260 GO TO 180
```

If the input is not recognized, go back and display the picture at the original location.

```
T: 270 K=RL$(I)
T: 280 GO TO 185
```

Set the displacement to the value corresponding to the direction that you want to move in. Go back to display the picture in the new location.

```
T: RUN
```

VI. SOME GRAPHICS HINTS AND KINKS

A. When is π not π ?

The character π has at least two different representations.

Exercise: Demonstrate the problem of having two different representations of .

```
T: NEW
T: 400 GET A$
T: 405 IF LEN(A$)=0 THEN 400
T: 410 IF A$ < > "  $\pi$  " THEN 900
T: RUN 400
T:  $\pi$ 
R: ?UNDEF 'D STATEMENT ERROR IN 410
R: READY.
T: ?ASC(A$),ASC("  $\pi$  ")
R: 222      255
R: READY.
T: ?A$
R:  $\pi$ 
R: READY.
T: ?CHR$(222),CHR$(255)
R:  $\pi$        $\pi$ 
R: READY.
```

This sequence shows that when π is entered in a program, it has the numeric value of 255. However, when π is entered as input, it has the numeric value of 222.

B. What happens to Reverse Video?

The PET keyboard has a key marked RVS ON (reverse video screen) and RVS OFF (turn off reverse video). Use of that key has some strange results.

Exercise: Demonstrate the affect of RVS ON and RVS OFF.

```
T: NEW
T: 'CLR'
T: 100 READ A$,B$,C$
T: 110 DATA 'RVS ON' [A], [B], [C] (A,B,C appear as reverse video)
T: 120 PRINT A$;B$;C$
T: RUN
R: ABC (normal, i.e., not reverse video)
R: READY.
```

PET GRAPHICS

```
T: LIST
R: 100 READ A$,B$,C$
R: 110 DATA A,B,C, (normal, i.e., not reverse video!)
R: 120 PRINT A$;B$;C$
R: READY.
T: 110 DATA "'RVS ON'A",'RVS ON'B",'RVS ON'C"
T: RUN
R: ABC (reverse video!)
R: READY.
```

To enter reverse video characters you must not enter reverse video characters! You must type 'RVS ON' within the quote string, i.e., "'RVS ON'X" will give the correct results; 'RVS ON'"X" will not.

VII. APPENDICES

APPENDIX A - PROGRAM LISTINGS

Complete Sequential List of Character Table.

This program displays all the PET's symbols in a table format. You can determine the numeric value of any character by multiplying the column by 10 and adding the row. The program that starts at line 300 allows you to check the symbol displayed for a given numeric value.

```
90 PRINT "GRAPHIC OR LOWER CASE"
92 INPUT A$
94 IF A$="GRAPHICS" THEN POKE 59468,12
96 IF A$="LOWER" THEN POKE 59468,14
99 PRINT "'CLR'"
100 PRINT "CHARACTER DISPLAY TABLE (POKE)"
110 PRINT "    00          10          20"
120 PRINT "    01234567890123456789012345"
125 Q=32891
130 FOR J=0 TO 9
140 PRINT J;SPC(26);J
150 NEXT J
152 PRINT
154 PRINT "    00          10          20"
156 PRINT "    01234567890123456789012345"
160 FOR J=0 TO 9
170 FOR I=0 TO 24
180 POKE Q+((J+1)*40+I),I*10+J
190 NEXT I
210 NEXT J
215 I=25
220 FOR J=0 TO 5
230 POKE Q+((J+1)*40+I),I*10+J
240 NEXT J
300 INPUT K
305 IF K < 0 OR K > 255 THEN 315
310 POKE 32848,K
315 PRINT "↑↑↑ "
320 GO TO 300
```

PET GRAPHICS

Low-density Plotter Listing.

This program displays an X,Y plot of the sine function on a 40x25 grid.

✓

```
100 PT%=ASC("*")
103 PRINT "'CLR' PLOT OF SINE 0 TO 2 "
104 XL=0:XB=2.52* $\pi$ 
105 YLYL=-1.2:YB=1.2
106 SS=.1
107 FOR I=XL TO XB STEP SS
108 X=I
110 Y=SIN(X)
120 GO SUB 1000
130 NEXT I
140 GO TO 103
1000 XS%=39*((X-XL)/(XB-XL))
1010 YS%=24*((YB-Y)/(YB-YL))
1050 Q=YS%*40+XS%+32768
1310 POKE Q,PT%
1320 RETURN
```

*uitbreiden met k-as
y-as*

APPENDICES

High-density Plotter Listing.

This program displays an X,Y plot of the sine function on an 80x50 grid.

with 6 rows per x-axis and y-axis

```

90 DIM TB(15)
91 FOR I=0 TO 15
92 READ TB(I)
94 DATA 32, 123, 126, 97, 108, 98, 127, 252, 124
    , 255, 226, 236, 225, 254, 251, 160
96 NEXT I
103 PRINT "'CLR' PLOT SINE 0 TO 2  $\pi$  "
104 XL=0:XB=122 2* $\pi$ 
105 YL=-1.2:YB=1.2
106 SS=.05
107 FOR I=XL TO XB STEP SS
108 X=I
110 Y=SIN(X)
120 GO SUB 1000
130 NEXT I
140 GO TO 103
1000 XS%=79*((X-XL)/(XB-XL))
1010 YS%=49*((YB-Y)/(YB-YL))
1020 XC%=XS%/2:YL%=YS%/2
1050 Q=YL%*40+XC%+32768
1060 I%=XS% AND 1
1062 J%=(YS%+1) AND 1
1064 K%=(J%+1)*(I%*3+1)
1070 K=PEEK(Q)
1080 FOR J=0 TO 15
1090 IF K=TB(J) THEN 1300
1100 NEXT J
1110 J=0
1300 K%=K% OR J
1308 PT%=TB(K%)
1310 POKE Q,PT%
1320 RETURN

```

PET GRAPHICS

Complete Sequential List of SKETCH.

koppelen aan volgende?

SKETCH lets you draw on the PET TV display with the keyboard. You can save on cassette what you have entered or display sketches previously saved.

```

100 DIM PIC%(255),LOC(255)
110 A$="'CLR' ↓↓↓↓"
115 PRINT "←b←";A$;"#";
120 GET A$
130 IF LEN(A$)=0 THEN 120
140 IF A$<>"↑" THEN 115
150 K%=0
155 PRINT "← b"
160 FOR I=32768 TO 33767
170 K=PEEK(I)
180 IF K=32 THEN 220
190 K%=K%+1
200 IF K% > 255 THEN 900
210 PIC%(K%)=K
215 LOC(K%)=I
220 NEXT I
225 PRINT "'HOME'"
230 PRINT "WHAT NEXT?"
240 INPUT A$
250 IF A$="W" THEN 400
255 IF A$="N" THEN 110
257 IF A$="R" THEN 500
260 IF A$<>"D" THEN 225
270 PRINT "'CLR'"
280 FOR I=1 TO K%
290 POKE LOC(I),PIC%(I)
300 NEXT I
310 GO TO 230
400 OPEN 1,1,2,"PICTURE"
405 FOR I=1 TO K%
410 T=TI
420 PRINT #1,STR$(LOC(I));",";STR$(PIC%(I))
430 IF TI-T < 120 THEN 470
440 POKE 59411,53:T=TI
450 IF TI-T < 6 THEN 450
460 POKE 59411,61
470 NEXT I
475 PRINT #1,STR(0);",";STR(0)
476 CLOSE 1
478 PRINT "WRITE DONE" "HOME" write done 'home'
480 GO TO 230
500 OPEN 1,1,0,"PICTURE"
510 PRINT "'CLR' ↓↓↓↓"
520 INPUT #1,A,B
530 IF A=0 AND B=0 THEN 560

```

Bij move:
↑ } ⇒ opslag tekening.
D. }
dan move + evt aanvullen.
ook problemen moves
op te laten staan zodat
tekening niet zichzelf move't.

invullen
aanvullen +
verplaatsen

for X = i + j.
290 POKE LOC(X),PIC%(I)

Alles via D.
kgk eerst of
aanvullen
nu ook mogelijk
in mbu M.

APPENDICES

```
540 POKE A,B
550 GO TO 520
560 CLOSE 1
565 A$="b"
570 GO TO 110
900 PRINT "TOO MANY SYMBOLS"
910 K%=255:GO TO 230
```

PET GRAPHICS

Complete Sequential List of MOVER.

MOVER lets you move a sketch one step in any of eight directions from its original position.

*koppelen aan wijze-
moeder - proberen met cursor-
en get.*

```
100 FOR I=1 TO 9
110 READ N$(I),RL%(I)
120 DATA "UL",-41,"U",-40,"UR",-39,"R",1
    ,"DR",41,"D",40,"DL",39,"L",-1,"B",0
130 NEXT I
140 FOR I=1 TO 4
150 READ LOC(I),PIC%(I)
160 DATA 33267, 79, 33268, 80, 33307, 76, 3330
    8, 122
170 NEXT I
180 K=0
185 PRINT "'CLR'";
190 FOR I=1 TO 4
200 POKE LOC(I)+K,PIC%(I)
210 NEXT I
220 INPUT DR$
230 FOR I=1 TO 9
240 IF DR$=N$(I) THEN 270
250 NEXT I
260 GO TO 180
270 K=RL%(I)
280 GO TO 185
```

APPENDIX B - CHARACTER SETS

The following character sets are those used by the Commodore PET. The character sets described will be those used with the poke statement and those used with the PRINT, CHR\$, and ASC statements. There are three sets for the POKE statement, graphics, lower case, and reverse video. There are only two sets for the PRINT, CHR\$, and ASC statements. One is for graphics and the other is for lower case.

The character set used with the PRINT statement is not as straight forward as the one used with the POKE statement. For example, to get a reverse video character with the PRINT you must first hit the double quote ("), then the 'RVS' key. All characters after that will be reverse video until the terminating double quote (").

Example of displaying reverse video characters.

```
T: PRINT "'RVS ON'AB"
R: AB
R: READY.
```

The characters in boxes represent reverse video characters. The R is the character you get when the reverse video key is typed.

An alternative method to get reverse video characters is to type CHR\$(18) rather than type the 'RVS ON' key.

Example of displaying reverse video characters.

```
T: PRINT CHR$(18);"ABC"
R: ABC
R: READY.
```

Some characters have a dual numeric representation when used with the CHR\$ function. For example, CHR\$(35)=CHR\$(99)=#. The following is a list of those numbers that represent the same character.

33,97	47,111	60,124	170,234	183,247
34,98	48,112	61,125	171,235	184,248
36,100	49,113	62,126	172,236	185,249
37,101	50,114	63,127	173,237	186,250
38,102	51,115	161,225	174,238	187,251
39,103	52,116	162,226	175,239	188,252
40,104	53,117	163,227	176,240	189,253
41,105	54,118	164,228	177,241	190,254
42,106	55,119	165,229	178,242	222,255
43,107	56,120	166,230	179,243	
44,108	57,121	167,231	180,244	
45,109	58,122	168,232	181,245	
46,110	59,123	169,233	182,246	

PET GRAPHICS

A. Character Codes for the PRINT, CHR\$, and ASC.

Lower-case character set (location 59468=14).

161	162	163	164	165	167	166	220	168	169	223
33	34	35	36	37	39	38	92	40	41	95

	$=128+17$		$=128+17$
147	145	157	148
19	17	29	20

209	215	197	210	212	217	213	201	207	208	222
81	87	69	82	84	89	85	73	79	80	94

183	184	185	175
55	56	57	47

193	211	196	198	199	200	202	203	204	186
55	83	68	70	71	72	74	75	76	58

141

RETURN

180	181	182	170
52	56	57	42

218	216	195	214	194	206	205	172	187	191
90	88	67	86	66	78	77	14	59	63

177	178	179	171
49	50	51	43

146	192	219	221	160	188	190
18	64	91	93	32	60	62

176	174	173	189
48	46	45	61

APPENDICES

B. Character Codes for the PRINT, CHR\$, and ASC.

Graphics character set (location 59468=12).

161	162	163	164	165	167	166	220	168	169	223	147	145	157	148
33	34	35	36	37	39	38	92	40	41	95	19	17	29	20

209	215	197	210	212	217	213	201	207	208	222	183	184	185	175
81	87	69	82	84	89	85	73	79	80	94	55	56	57	47

193	211	196	198	199	200	202	203	204	186	180	181	182	170
65	83	68	70	71	72	74	75	76	58	52	53	54	42

218	216	195	214	194	206	205	172	187	191	177	178	179	171
90	88	67	86	66	78	77	44	59	63	49	50	51	43

146	192	219	221	160	188	190	141
18	64	91	93	32	60	62	13

176	174	173	189
48	46	45	61

PET GRAPHICS

C. Character Codes for the POKE Statement.

Reverse video lower-case character set (location 59468=14).

12 60
225 226 227 228 229 231 230 220 232 233 223

/	//	#	\$	%	'	&	\	(	)	←
---	----	---	----	---	---	---	---	---	---	---

161 162 163 164 165 167 166 156 168 169 159

209 215 197 210 212 217 213 201 207 208 222

Q	W	E	R	T	Y	U	I	O	P	↑
---	---	---	---	---	---	---	---	---	---	---

145 151 133 146 148 153 149 137 143 144 158

247 248 249 239

7	8	9	/
---	---	---	---

183 184 185 175

193 211 196 198 199 200 202 203 204 250

A	S	D	F	G	H	J	K	L	:
---	---	---	---	---	---	---	---	---	---

129 147 132 134 135 136 138 193 140 186

244 245 246 234

4	5	6	*
---	---	---	---

180 181 182 170

218 216 195 214 194 206 205 236 251 255

Z	X	C	V	B	N	M	,	;	?
---	---	---	---	---	---	---	---	---	---

154 152 131 150 130 142 141 172 187 191

241 242 243 235

1	2	3	+
---	---	---	---

177 178 179 171

192 219 221 224 252 254

SHIFT	OFF RVS ON	@	[	]	SPACE	<	>	RUN STOP	SHIFT
-------	------------------	---	---	---	-------	---	---	-------------	-------

128 155 157 160 188 190

240 238 237 253

0	.	-	=
---	---	---	---

176 174 173 189

15

APPENDICES

D. Character Codes for the POKE Statement.

Graphics character set (location 59468=12).

3 5

97	98	99	100	101	103	102	92	104	105	95
33	34	35	36	37	39	38	28	40	41	31

81	87	69	82	84	89	85	73	79	80	94
17	23	5	18	20	25	21	9	15	16	30

119	120	121	111
55	56	57	47

65	83	68	70	71	72	74	75	76	122
1	19	4	6	7	8	10	11	12	58

116	117	118	106
52	53	54	42

90	88	67	86	66	78	77	108	123	127
26	24	3	22	2	14	13	44	59	63

113	114	115	107
49	50	51	43

	64	91	93	96	124	126		
18	0	27	29	32	60	62		

112	110	109	125
48	46	45	61

RETURN

PET GRAPHICS

E. Character Codes for the POKE Statement.

Lower-case character set (location 59468=14).

97	98	99	100	101	103	102	92	104	105	95
33	34	35	36	37	39	38	28	40	41	31

81	87	69	82	84	89	85	73	79	80	94
17	23	5	18	20	25	21	9	15	16	30

119	120	121	111
55	56	57	47

65	83	68	70	71	72	74	75	76	122
1	19	4	6	7	8	10	11	12	58

116	117	118	106
52	53	54	42

90	88	67	86	66	78	77	108	123	127
26	24	3	22	2	14	13	44	59	63

113	114	115	107
49	50	51	43

RETURN

	64	91	93	96	124	126	
	0	27	29	32	60	62	

112	110	109	125
48	46	45	61

Notes

AND 4-2 ; 4-0
or ; peek 4-5-
Ti 5-3
hog doorheme 4-0

Notes

Notes

Notes

Notes